On the relevance of Organizational Structures for a Technology of Agreement

Holger Billhardt, Roberto Centeno, Alberto Fernández,
Ramón Hermoso, Rubén Ortiz, Sascha Ossowski, Matteo Vasirani

Centre for Intelligent Information Technologies (CETINIA),
Universidad Rey Juan Carlos,
Calle Tulipán s/n,
28933 Móstoles (Madrid), Spain

Abstract. This paper provides a brief overview of the field of coordination in multi-agent systems, and outlines its relation to current efforts working towards a paradigm for smart, next-generation distributed systems, where coordination is based on the concept of agreement between computational entities. To illustrate the types of mechanisms that we envision to be part of a "technology of agreement", we provide two examples of how techniques from the field of organisations can be used to foster coordination and agreement in open multi-agent systems.

1 Introduction

An increasing number of transactions and interactions at business level, but also at leisure level, are nowadays mediated by computers and computer networks. An appealing way to model and design such applications is by purposefully combining components to which more and more complex tasks can be delegated. These components need to show an adequate level of intelligence, should be capable of sophisticated ways of interacting, and are usually massively distributed, sometimes embedded in all sort of appliances and sensors. In order to allow for an efficient design and implementation of systems of these characteristics, it is necessary to effectively enable, structure, and regulate their communications in different contexts.

Such an enterprise raises a number of technological challenges. Firstly, the open distributed nature of such systems adds to the *heterogeneity* of its components. The system structure may evolve at runtime, as new nodes may appear or disappear at will. There is also a need for on-the-fly alignment of certain concepts that interactions relate to, as the basic ontological conventions in such systems will be very limited. The *dynamicity* of the environment calls for a continuous *adaptation* of the structures that regulate the components' interactions, so as to achieve and sustain desired functional properties. But also non-functional issues related to *scalability*, *security*, and *usability* need to be taken into account. When designing mechanisms that address these challenges, the notion of *autonomy* becomes central: components may show complex patterns of activity aligned with the different goals of their designers, while it is usually impossible to directly influence their behaviour from the outside.

Coordination in multi-agent system (MAS) aims at harmonising the interactions of multiple autonomous components or agents. Therefore, it appears promising to review different conceptual frameworks for MAS coordination, and to analyse the potential and limitations of the work done in that field with regard to some of the aforementioned challenges.

This paper is organised as follows. Section 2 provides a brief overview of coordination in MAS, identifies the notion of *agreement* as a centrepiece of an integrated approach to coordination in open distributed systems, and outlines some research topics related to the vision of a technology of agreement. Section 3 provides examples of how organisational structures can be used to instil coordination and agreement in open multi-agent systems, in the realm of matchmaking and trust mechanisms. Some conclusions are drawn in Section 4.

2 Coordination and Agreement in Multi-agent Systems

Maybe the most widely accepted conceptualisation of coordination in the MAS field originates from Organisational Science. It defines coordination the *management of dependencies* between organisational activities [20]. In a multi-agent setting, the subjects whose activities need to be coordinated are the agents, while the entities between which dependencies are usually goals, actions or plans. Depending on the characteristics of the MAS environment, a taxonomy of dependencies can be established, and a set of potential coordination actions assigned to each of them (e.g. [36], [23]). Within this model, the *process* of coordination is to accomplish two major tasks: first, a *detection* of dependencies needs to be performed, and second, a *decision* respecting which coordination action to apply must be taken. A coordination *mechanism* shapes the way that agents perform these tasks [21].

From a *macro-level* (MAS-centric) perspective, the outcome of coordination can be conceived a “global” plan (or decision, action etc.). This may be a “joint plan” [28] if the agents reach an explicit agreement on it during the coordination process, or just the sum of the agents' individual plans (or decisions, actions etc. – sometimes called “multi-plan” [24]) as perceived by an external observer. Roughly speaking, the quality of the outcome of coordination at the macro-level can be evaluated with respect to the agents’ joint goals or the desired functionality of the MAS as a whole. If no such notion can be ascribed to the MAS, other, more basic features can be used instead. A good result of coordination, for instance, often relates to “efficiency”, which frequently comes down to the notion of Pareto-optimality. The amount of resources necessary for coordination (e.g. the number of messages necessary) is also sometimes used as a measure of efficiency.

The dependency model of coordination appears to be particularly adequate for *representing* relevant features of coordination problems in MAS. Frameworks based on this model have been used to capture coordination requirements in a variety of interesting MAS domains (e.g. [8]). Still, dependency detection may become a rather knowledge intensive task, which is further complicated by incomplete and potentially inconsistent local views of the agents. From a design perspective, coordination is probably best conceived as the effort of *governing the space of interaction* [5] of a MAS, as the basic challenge amounts to how to make agents converge on interaction patterns that adequately (i.e. instrumentally with respect to desired MAS features) solve the dependency detection and decision tasks. A variety of approaches that tackle this problem can be found in the literature, shaping the interaction space either directly, by making assumptions on agent behaviours and/or knowledge, or indirectly, by modifying the agent's environment [30] (e.g. the MAS infrastructure [22], or the institutional context [10]). The applicability of these mechanisms depends largely on the number and type of assumptions that one may make regarding the possibility of manipulating agent programs, agent populations, or the agents' environment. This, in turn, is dependent on the characteristics of the coordination problem at hand.

From the point of view of an individual agent, the problem of coordination boils down to finding the sequence of actions that, given the regulations within the system (or, if this possible in a certain environment, the expected cost of transgressing them), best achieves its goals. In practice, this implies a series of non-trivial problems. Models of coalition formation determine when and with whom to form a team for the achievement of some common (sub-) goal, and how to distribute the benefits of synergies that arise from this cooperation [32]. Distributed planning approaches [9] determine how to (re-)distribute tasks among team members and how to integrate results. From an individual agent's perspective, the level of trustworthiness of others is central to almost every stage of these processes, so as to determine whether other agents are likely to honour the commitments that have been generated [33].

Several quite different approaches and mechanisms coexist under the "umbrella" of the term coordination in MAS [25]. Not all of them are relevant to the challenges for the design of open distributed systems outlined in the introduction. For instance, the whole set of *coupled* coordination mechanisms [35] are effectively useless for the purpose of this paper, as they require having a direct influence on the agent programs. On the other hand, the problem of semantic interoperability is usually outside the scope of MAS coordination models and languages.

The notion of *agreement* among computational agents appears to be better suited as the fundamental notion for the proposal outlined in this paper. Following a recent research effort in the field of "Agreement Technologies" [1], the process of agreement-based coordination can be conceived based on two main elements:

(1) a normative context, that determines the rules of the game, i.e. interaction patterns and additional restrictions on agent behaviour; and

(2) a call-by-agreement interaction method, where an agreement for action between the agents that respects the normative context is established first; then actual enactment of the action is requested.

Methods and mechanisms from the fields of *semantic alignment*, *norms*, *organization*, *argumentation and negotiation*, as well as *trust and reputation* are envisioned be part of a "sandbox" to build software systems based on a technology of agreement [1].

Semantic technologies should constitute a centrepiece of such an enterprise as semantic problems pervade all the others. Solutions to semantic mismatches and alignment of ontologies [4] are needed to have a common understanding of norms or of deals, just to put two examples. As we will illustrate in the next section, the use of semantics-based approaches to service discovery and composition will allow exploring the space of possible interactions and, consequently, shaping the set of possible agreements [12].

At system-level, *norms* are needed to determine constraints that the agreements, and the processes to reach them, have to satisfy. Reasoning about a system's norms is necessary at design-time to assure that the system has adequate properties, but it may also be necessary at run-time, as complex systems usually need dynamic regulations [14]. *Organisational* structures further restrict the way agreements are reached by fixing the social structure of the agents: the capabilities of their roles and the relationships among them (e.g. power, authority) [3].

Moving further towards the agent-level, *negotiation* methods are essential to make agents reach agreements that respect the constraints imposed by norms and organisations. These methods need to be complemented by an argumentation-based approach: by exchanging arguments, the agents' mental states may evolve and, consequently, the status of offers may change [2] [6]. Finally, agents will need to use *trust* mechanisms that summarise the history of agreements and subsequent agreement executions in order to build long-term relationships between the agents [34].

Of course, these methods should not be seen in isolation, as they may well benefit from each other. For instance, in certain situations trust mechanisms may take advantage of the roles structures included in an organisational model, so as to improve their performance when only limited information about previous interactions is available.

3 Organizational structures and agreement

This section intends to illustrate the types of mechanisms that we envision being part of the agreement technology "sandbox" mentioned previously. In particular, we will provide examples of how organisational structures can be used to foster coordination and agreement in open MAS.

Organisational models underlying approaches such as Agent-Group-Role [11], MOISE [16], or RICA [31] provide a rich set of concepts to specify and structure mechanisms that govern agent interactions through the corresponding infrastructures or middleware. A key notion in most organisational models is the concept of role. Roles can often be organised in a taxonomy, which can be modelled as a pair $<R, \leq>$ where R is the set of concepts representing roles and $\leq$ is a partial order among R.

In section 3.1, we show how role taxonomies can be used to locate suitable interactions partners, by providing additional information regarding the usability of services in a certain interaction context. Section 3.2 outlines how such taxonomies can be used for the bootstrapping of reputation mechanisms, when only limited information about past interactions is available in the system.

3.1 Organisational structures and matchmaking mechanisms

Our first example refers to service-oriented MAS where the capabilities of agents are modelled in the shape of services which, in turn, are described by some standard service description language. In the following we present our approach to enriching service descriptions with organisational information. For this purpose, we first introduce simple languages for representing role-based service advertisements and service requests.

A service advertisement S is a set of pairs so that

$$S \subseteq \left\{ < r, \rho > \mid r \in R, \rho = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} r_{ij}, r_{ij} \in R \right\}$$

In this definition, r is the role played by the provider in the interaction, and ρ is a set of roles that must be played by the requester agent for the correct accomplishment of the service, given by a formula in disjunctive normal form (DNF).

A service request Q is a set of pairs so that

$$Q = < \rho, C >, \rho = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} r_{ij}, r_{ij} \in R, C \subseteq R$$

Again, ρ is a DNF role expression (usually atomic) specifying the searched provider roles, and C is a set of roles that define the *capabilities* of the requester (the roles it is able to play).

Although organisational information is not a first-class citizen in service description languages such as OWL-S[1] or WSMO[2], it is not difficult to incorporate it into them. In OWL-S, for instance, we propose to include the role description as an additional parameter, called *Service_Roles*, in the case of service descriptions (r and ρ are mapped to *providerRole* and *dependingRoles* tags, respectively), and *Query_Roles* for service requests (ρ and C are mapped to *SearchedProviderRoles* and *CapabilityRoles*) [12].

In many multi-agent settings, this kind of organisational information can be used to complement standard I/O based matchmaking in order to improve its performance. We set out from the following requirements to define a semantic match function between two roles:

1. It must return a real number in the range [0..1] (degree of match or *dom*), with a higher value the more similar the concepts are (1 if $r_1=r_2$).
2. It must consider the distance between both concepts (roles) in the ontology: the greater the distance, the less similar are the concepts (decreasing function).
3. The change of *dom* per unit must decrease inversely with the distance (e.g., the step from 1 to 2 is more relevant than 5 to 6).
4. The $dom(r_1,r_2)$ must be independent of the height of the taxonomy and its location within it.
5. The logical relation between the two roles (i.e. the subsumption relation) must be taken care of. This is the most important criterion to take into account.

---

[1] http://www.daml.org/services/owl-s/

[2] http://www.wsmo.org/

Requirement 2 is addressed by using the measure proposed by Rada [27], consisting of the number of edges in the shortest path between two concepts in the taxonomy:

$$dist(c_1,c_2) = depth(c_1) + depth(c_2) - 2 \times depth(lcs^3(c_1,c_2))$$

Requirement 3 imposes a non-linear decreasing function. We use a typical exponential function here, $e^{-dist(r_1,r_2)}$, as it maintains its range in [0..1], is monotonically decreasing, is 1 when $r_1=r_2$ (requirement 1), and it does neither depend on the height of the taxonomy nor on the global height of the roles (requirement 4).

In order to comply with requirement 5, we differentiate among the four levels of match proposed by Paolucci et al. [26] (advertisement *A* and request *Q*):

- *exact*: if $r_A = r_Q$
- *plug-in:* if r_A *subsumes* r_Q
- *subsumes*: if r_Q *subsumes* r_A
- *fail*: otherwise

We take the final value, representing the degree of match, equal to 1 in case of an *exact* match, it varies between 1 and 0.5 in case of a *plug-in* match, stays between 0.5 and 0 in case of a *subsumes* match, and it is equal to 0 in case of a *fail*. So we only have to scale the value [0..1] to the ranges [0..0.5] and [0.5..1].

Based on these considerations, we define the degree of matching *dom* between two roles R_A and R_Q as

$$dom(R_A, R_Q) = \begin{cases} 1 & if\ R_A = R_Q \\ \frac{1}{2} + \frac{1}{2 \cdot e^{\|R_A, R_Q\|}} & if\ R_A\ is\ subclass\ of\ R_Q \\ \frac{1}{2} \cdot e^{\|R_A, R_Q\|} & if\ R_Q\ is\ subclass\ of\ R_A \\ 0 & otherwise \end{cases}$$

where $\|R_A,R_Q\|$ is the distance between R_A and R_Q (*dist*(R_A,R_Q)) in the role taxonomy (if there is a subsumption relation between them). By construction, this equation fits the requirements.

The *semantic match* between a service advertisement *S* and a query *Q* (service request) is done by searching the role in *S* that best matches the one in *Q*. The degree of match between a role in the request and a service advertisement, given the set of capabilities of the requester, is done by comparing the searched role with every other given role and returns the maximum degree of match. For each role in the advertisement, the match between the provider roles is made, as well as the match between the depending roles and the capabilities of the requester.

The minimum of both values is considered the degree of match. In case of logical expressions, the minimum is used as combination function for the values in a conjunction and the maximum for disjunctions (which always keep the value resulting of the combination within the range [0,1]). Details of the algorithm used to determine the degree of match between a service request and a service advertisement are described in [12].

---

[3] Least common subsumer

Our approach is intended to be complementary to other general-purpose matchmakers. We have performed experiments combining an implementation of the semantic match between services (ROWLS) with OWLS-MX [19], one of the leading hybrid matchmakers available to-date. Comparing a combination of ROWLS and OWL-MX to a standalone use of the latter, we have found an improvement to both effectiveness and efficiency based on our test collection [12].

3.2 Organisational structures and trust mechanisms

The second example shows how an agent can use knowledge about the organisational structure to infer confidence in a situation when no previous experience about a specific interaction is available. Similar to other approaches [17][29], we set out from a trust model based on the idea of *confidence* and *reputation*. Both ratings evaluate the trustworthiness of other agents in a particular situation (e.g., playing a particular role in a particular interaction). *Confidence* is a local measure that is only based on an agent's own experiences, while *reputation* is an aggregated value an agent gathers by asking its acquaintances about their opinion regarding the trustworthiness of another agent. Thus, reputation can be considered as an external *social* measure. We define *trust* as a rating resulting from combining *confidence* and *reputation* values.

A typical scenario for the use of a trust model is the following. An agent A wants to evaluate the trustworthiness of some other agent B -- playing the role R -- in the interaction I. This trustworthiness is denoted as $t_{A\to\langle B,R,I\rangle} \in [0..1]$, measuring the trust of A in B (playing role R) being a "good" counterpart in the interaction I. When evaluating the trustworthiness of a potential counterpart, an agent can combine its local information (confidence) with the information obtained from other agents regarding the same counterpart (reputation).

Confidence, $c_{A\to\langle B,R,I\rangle}$, is collected from A's past interactions with agent B playing role R and performing interactions of type I. We call *Local Interaction Table* (LIT) an agent's data structure storing confidence values for past interactions with any counterpart the agent has interacted with. Each entry corresponds to a *situation*: an *agent* playing a specific *role* in a particular *interaction*. LIT_A denotes agent A's LIT. An example is shown in Table 1. Each entry in a LIT consists of: (i) the Agent/Role/Interaction identifier <*X,Y,Z*>, (ii) the confidence value for the issue ($c_{A\to\langle X,Y,Z\rangle}$), and (iii) a reliability value ($r_{A\to\langle X,Y,Z\rangle}$). The confidence value is obtained from some function that evaluates past experiences on the same situation. We suppose $c_{A\to\langle X,Y,Z\rangle} \in [0..1]$ where higher values represent higher confidence.

$\langle X,Y,Z\rangle$	$c_{A\to\langle X,Y,Z\rangle}$	$r_{A\to\langle X,Y,Z\rangle}$
$\langle a_9,r_2,i_3\rangle$	0.2	0.75
$\langle a_2,r_7,i_1\rangle$	0.7	0.3
⋮	⋮	⋮
$\langle a_9,r_2,i_5\rangle$	0.3	0.5

Table 1. An agent’s local interaction table (LIT_A)

Each direct experience of an agent regarding a situation <*X,Y,Z*> changes its confidence value $c_{A\to\langle X,Y,Z\rangle}$. In this sense, we suppose that the agents have some mechanism to evaluate the behaviour of other agents that they interact with. Let $g_{\langle X,Y,Z\rangle} \in [0..1]$ denote the evaluation value an agent *A* calculates for a particular experience with the agent *X* playing role *Y* in the interaction of type *Z*. We use the following formula to update confidence:

$$c_{A\to\langle X,Y,Z\rangle} = \varepsilon \cdot c'_{A\to\langle X,Y,Z\rangle} + (1-\varepsilon)\cdot g_{\langle X,Y,Z\rangle}$$

where $c'_{A\to\langle X,Y,Z\rangle}$ is the confidence value in *A*'s LIT before the interaction is performed and $\varepsilon \in [0..1]$ is a parameter specifying the importance given to *A*'s past confidence value. In general, the aggregated confidence value from past experiences will be more relevant than the evaluations of the most recent interactions.

Reliability ($r_{A\to\langle X,Y,Z\rangle}$) measures how certain an agent is about its own confidence in a situation. We suppose $r_{A\to\langle X,Y,Z\rangle} \in [0..1]$. Furthermore, we assume that $r_{A\to\langle X,Y,Z\rangle} = 0$ for any tuple <*X,Y,Z*> not belonging to LIT_A. We calculate reliability by using the approach proposed by Huynh, Jennings and Shadbolt [18], taking into account the number of interactions a confidence value is based on and the variability of the individual values across past experiences.

An agent may build trust directly from its confidence value or it may combine confidence with reputation. Reputation is particularly useful when an agent has no experience or if the reliability value for the confidence is not high. Social reputation may be obtained by asking other agents about their opinion on a situation. Agents that have been requested for their opinion will return the corresponding confidence and reliability ratings from their LIT. The requester might then be able to build trust by calculating a weighted mean over its own confidence value and the confidence values received from others, as it is represented in the following equation:

$$t_{A\to\langle B,R,I\rangle} = \begin{cases} c_{A\to\langle B,R,I\rangle} & \text{if } r_{A\to\langle B,R,I\rangle} > \theta \\ \dfrac{\sum_{x\in AA\cup\{A\}} c_{x\to\langle B,R,I\rangle}\cdot\omega_{x\to\langle B,R,I\rangle}}{\sum_{x\in AA\cup\{A\}} \omega_{x\to\langle B,R,I\rangle}} & \text{otherwise} \end{cases}$$

$\theta \in [0..1]$ is a threshold on the reliability of confidence. If the reliability is above θ then an agent’s own confidence in a situation is used as the trust value. Otherwise trust is build by combining confidence and reputation. *AA* is a set of acquaintances an agent asks about their opinion regarding the situation <*B,R,I*>. For instance, in some scenarios it may be useful to ask other agents that play the same role as *A*, since they may have similar interests and goals.

The weights $\omega_{x\to\langle B,R,I\rangle}$ given to the gathered confidence values are composed of the corresponding reliability values and a constant factor α that specifies the importance given to *A*'s own confidence in the issue:

$$\omega_{x\to\langle B,R,I\rangle} = \begin{cases} r_{x\to\langle B,R,I\rangle}\cdot\alpha & \text{if x = A} \\ r_{x\to\langle B,R,I\rangle}\cdot(1-\alpha) & \text{otherwise} \end{cases}$$

Basic trust models as the one outlined before run into problems when no interactions of a specific type have been performed before and, in addition, social reputation is not available or not reliable. In such a situation, information of the organisational structure can be used to determine an approximate degree of trust.

In particular, one approach consists of using the agent/role confidence $c_{A\to\langle B,R,\_\rangle}$ (or the agent confidence $c_{A\to\langle B,\_,\_\rangle}$) as an estimation for $c_{A\to\langle B,R,I\rangle}$ if agent A has no reliable experience about situation <*B,R,I*>. This approach relies on the hypothesis that, in general, *agents behave in a similar way in all interactions related to the same role*. We argue that, exploiting this idea, the more similar I' and I are, the more similar the values $c_{A\to\langle B,R,I'\rangle}$ and $c_{A\to\langle B,R,I\rangle}$ will be. The same applies to roles.

Taking this assumption further, confidence ratings for similar agent/role/interaction tuples can be accumulated to provide evidence for the trustworthiness of the situation <*B,R,I*>. Based on this idea, we propose to build trust by taking into account all the past experiences an agent has, focusing on their degree of similarity between organisational concepts, with the situation <*B,R,I*>. In particular, we calculate trust as a weighted mean over all the confidence values an agent has accumulated in its LIT. This is shown in the following equation:

$$t_{A\to\langle B,R,I\rangle} = \frac{\sum_{\langle X,Y,Z\rangle\in LIT_A} c_{A\to\langle X,Y,Z\rangle}\cdot\omega_{A\to\langle X,Y,Z\rangle}}{\sum_{\langle X,Y,Z\rangle}\omega_{A\to\langle X,Y,Z\rangle}}$$

$\omega_{A\to\langle X,Y,Z\rangle}$ is the weight given to agent A's confidence on situation <*X,Y,Z*>. The weights combine the confidence reliability with the similarity of the situation <*X,Y,Z*> to the target issue <*B,R,I*> in the following way:

$$\omega_{A\to\langle X,Y,Z\rangle} = r_{A\to\langle X,Y,Z\rangle}\cdot sim(\langle X,Y,Z\rangle,\langle B,R,I\rangle)$$

The similarity function $sim(\langle X,Y,Z\rangle,\langle B,R,I\rangle)$ is computed as the weighted sum of the similarities of the individual elements (agent, role and interaction) as it is shown in the following equation:

$$sim(\langle X,Y,Z\rangle,\langle B,R,I\rangle) = \begin{cases}\beta\cdot sim_R(R,Y)+\gamma\cdot sim_I(I,Z) & \textit{if}\ \mathrm{X}=\mathrm{B}\\ 0 & \textit{otherwise}\end{cases}$$

where $sim_R(R,Y), sim_I(I,Z)\in[0..1]$ measures the similarity between roles and interactions, respectively, and β and γ with $\beta+\gamma=1$, are parameters specifying the sensibility regarding the individual similarities.

Role similarities can be inferred from role taxonomies contained in an organisational model. In particular $sim_R(R,R')$ and can rely on a *distance function*, similar to the one presented in the previous subsection, that estimates the similarity between two roles on the basis of their proximity in the taxonomy. The same holds for $sim_I(I,I')$ when an interaction taxonomy is available in an organisational model [15].

Especially if an agent has no reliable experience about a particular agent/role/interaction situation, our organisation-based approach can be used to estimate trust without the necessity to rely on the opinions of other agents. So, role and interaction taxonomies can help making agents that use trust mechanisms less vulnerable to dishonest counterparts, as there is less need to rely on third-party information.

4 Discussion

This paper has presented an overview of different approaches to coordination in the MAS field. It has been argued that the notion of agreement is essential to instil coordination in open distributed systems. Some existing technologies from the field of MAS coordination can be applied to this respect, but others − and in particular semantic technologies − need to be added. To illustrate the types of mechanisms that we envision to be part of a "technology of agreement", we have provided two examples of how techniques from the field of organisations can be used to foster coordination and agreement in open MAS.

We have shown how organisational structures can be used to complement traditional matchmaking mechanisms so as to enhance their performance. We are currently evaluating as to how far more fine-grained quantitative matching techniques can be applied to this respect [13]. Furthermore, we have argued that organisational structures can be used to improve reputation mechanisms in situations where only a limited amount of information regarding previous interactions is available. Current work focuses on how, in turn, the history of interactions can be used to evolve organisational structures [15].

Several research efforts are currently ongoing that may contribute to the development of a "technology of agreement" in one or another way. The attempt to harmonise these efforts, which is currently being carried out at European level, promotes the emergence of a new paradigm for next generation distributed systems based on the notion of *agreement* between computational agents [7].

Acknowledgements

Some ideas reported in this paper draw upon joint with our partners in the framework of a Spanish national project on "Agreement Technology". This work was partially supported by the Autonomous Region of Madrid, grant URJC-CM-2006-CET-0300, and by the Spanish Ministry of Science and Innovation, grants TIN2006-14630-C03-02 and CSD2007-00022 (CONSOLIDER-INGENIO 2010).

References

[1] Agreement Technologies project homepage. http://www.agreement-technologies.org/

[2] Amgoud, L.; Dimopolous, Y.; Moraitis,P. (2007) : A unified and general framework for argumentation-based negotiation. Proc. 6th Int. Joint Conference on Autonomous Agents and Multi-Agents Systems, (AAMAS'2007). IFAAMAS, pages 963-970

[3] Argente, E.; Julian, V.; and Botti, V. (2006): Multi-Agent System Development based on Organizations. Elec. Notes in Theoretical Computer Science Vol. 150 No. 3, pages 55-71

[4] Atienza, M.; Schorlemmer, M. (2008): I-SSA – Interaction-situated Semantic Alignment. To appear in: Proc Int. Conf. on Cooperative Information Systems (CoopIS-2008)

[5] Busi, N.; Ciancarini, P.; Gorrieri, R.; Zavattaro, G. (2001): Coordination Models – A Guided Tour. Coordination of Internet Agents: Models, Technologies, and Applications (Omicini et al., eds.). Springer-Verlag, pages 6–24

[6] Caminada, M.; Amgoud, L. (2007): On the evaluation of argumentation formalisms. Artificial Intelligence Journal, V.171 (5-6). Elsevier, pages 286-310.

[7] COST Act. IC0801. http://www.cost.esf.org/index.php?id=110&action_number=IC0801

[8] Decker, K. (1996). TAEMS: A Framework for Environment Centered Analysis and Design of Coordination Mechanisms. Foundations of Distributed Artificial Intelligence (O'Hare and Jennings, eds.). John Wiley and Sons

[9] Durfee, E. (2001): Distributed Problem Solving and Planning. Multiagent Systems and Applications (Luck y otros, editores), LNAI 2086. Springer, pages 118-149

[10] Esteva, M; Rosell, B.; Rodríguez-Aguilar, J.A.; Arcos, J.Ll. (2004): AMELI – An agent-based middleware for electronic institutions. Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004). ACM Press, pages 236-243

[11] Ferber, J.; Gutknecht, O.; Fabien, M. (2003): From Agents to Organizations – An Organizational View of Multi-agent Systems. Agent-Oriented Software Engineering IV (Giorgini, Müller, AND Odell, eds.), LNCS 2935. Springer, pages 214-230

[12] Fernández, A.; Ossowski, S. (2008). Exploiting Organisational Information for Service Coordination in Multiagent Systems. Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2008). IFAAMAS, pages 257-264

[13] Fernández, A.; Polleres, A.; Ossowski, S. (2007). Towards Fine-grained Service Matchmaking by Using Concept Similarity. Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2) (Di Noia et al., eds.). CEUR Workshop Proceedings Vol. 243, pages 31-34

[14] Gaertner, D.; García-Camino, A.; Noriega, P.; Rodríguez-Aguilar, J.A.; Vasconcelos, W. (2007): Distributed norm management in regulated multiagent systems. Proc. Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007). IFAAMAS, pages 624-631

[15] Hermoso, R.; Centeno, R.; Billhardt, H.; Ossowski, S. (2008). Extending Virtual Organizations to improve trust mechanisms (Short Paper). Proc. Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2008). IFAAMAS, pages 1489-1492

[16] Hubner, J.; Sichman, J.; Boissier , O. (2006). Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. Int. Journal of Agent-Oriented Software Engineering, Vol.1, No.3/4. Inderscience, pages 370-395

[17] Huynh, T.D.; Jennings, N.R.; Shadbolt N. (2004) FIRE: An integrated trust and reputation model for open multi-agent systems. In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI), pages 18-22

[18] Huynh, T.D.; Jennings, N.R.; Shadbolt N.R. (2004) Developing an integrated trust and reputation model for open multi-agent systems. In Proceedings of 7th International Workshop on Trust in Agent Societies (AAMAS) , pages 65-74

[19] Klusch, M.; Fries, B.; Sycara, K. (2006): Automated Semantic Web Service Discovery with OWLS-MX. Proc. Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS-2006). ACM Press, pages 915-922

[20] Malone, T., Crowston, K. (1994): The Interdisciplinary Study of Co-ordination. Computing Surveys 26 (1). ACM Press, pages 87–119

[21] Omicini, A.; Ossowski, S. (2003): Objective versus Subjective Coordination in the Engineering of Agent Systems. Intelligent Information Agents – The European AgentLink Perspective (Klusch et al, eds). Springer-Verlag, pages 179–202

[22] Omicini, A.; Ossowski, S.; Ricci, A. (2004): Coordination Infrastructures in the Engineering of Multiagent Systems. Methodologies and software engineering for agent systems – The Agent-Oriented Software Engineering Handbook (Bergenti, Gleizes, and Zambonelli, eds.). Kluwer, pages 273-296

[23] Ossowski, S. (1999). Co-ordination in Artificial Agent Societies, LNAI 1535. Springer.

[24] Ossowski, S. (2001). Constraint Based Coordination of Autonomous Agents. Electronic Notes in Theoretical Computer Science 48. Elsevier, pages 211-226

[25] Ossowski, S.; Menezes, R. (2006): On Coordination and its Significance to Distributed and Multi-Agent Systems. Journal of Concurrency and Computation – Practice and Experience 18(4). Wiley, pages 359-370

[26] Paolucci, M., Kawamura, T., Payne, T., and Sycara, K. (2002): Semantic matching of web services capabilities. In Proceedings of the First International Semantic Web Conference on The Semantic Web, Springer-Verlag. pages 333-34

[27] Rada, R.; Mili, H.; Bicknell, E.; Blettner, M. (1989). Development and application of a metric on semantic nets. IEEE Trans. on Systems, Man and Cybernetics 19(1), pp. 17–30

[28] Rosenschein, J.; Zlotkin, G. (1995): Designing Conventions for Automated Negotiation. AI Magazine 15 (3), pages 29–46

[29] Sabater, J.; Sierra, C. REGRET: a reputation model for gregarious societies. (2001): In proceedings of the Fifth International Conference on Autonomous Agents (AGENTS). ACM Press, pages 194–195

[30] Schumacher, M.; Ossowski, S. (2006): The governing environment. Environments for Multiagent Systems II (Weyns et al., eds.). LNCS 3830. Springer, pages 88-104

[31] Serrano, J.M.; Ossowski, S. (2004): On the Impact of Agent Communication Languages on the Implementation of Agent Systems. Cooperative Information Agents VIII (Klusch et al., eds.). LNCS 2782. Springer, pages 92–106

[32] Shehory, O., Sycara, K. y Somesh, J. (1998): Multi-agent Coordination through Coalition Formation. Intelligent Agents IV – Agent Theories, Architectures and Languages (Singh, Rao, and Wooldridge). LNCS 1365. Springer, pages 143-154

[33] Sabater, J.; Sierra, C. (2005): Review on Computational Trust and Reputation Models. Artificial. Intelligence Review 24(1). Springer, pages 33-60

[34] Sierra, C.; Debenham, J. (2007): Information-Based Agency. Proc Int. Joint Conference on AI (IJCAI-2007). AAAI Press, pages 1513-1518.

[35] Tolksdorf, R. (2000): Models of Coordination. Engineering Societies in an Agent World (Omicini, Zambonelli, Tolksdorf, eds.). Springer-Verlag

[36] Von Martial, F. (1992): Co-ordinating Plans of Autonomous Agents. LNAI 610, Springer